AFRL-IF-RS-TR-2005-210
Final Technical Report
May 2005

AEROSPACE TECHNICAL SUPPORT FOR DARPA NETWORK MODELING AND SIMULATION PROGRAM AND COGNITIVE NETWORKS

University of Southern California

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. P771

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-210 has been reviewed and is approved for publication

APPROVED: /s/
WAYNE BOSCO
Project Engineer

FOR THE DIRECTOR: /s/
WARREN H. DEBANY, JR.
Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** May 2005	**3. REPORT TYPE AND DATES COVERED** **Final**	 **Jun 03 – Dec 04**

4. TITLE AND SUBTITLE

AEROSPACE TECHNICAL SUPPORT FOR DARPA NETWORK MODELING AND SIMULATION PROGRAM AND COGNITIVE NETWORKS

5. FUNDING NUMBERS

G - F30602-03-2-0135
PE - 62301E
PR - P771
TA - NM
WU - S1

6. AUTHOR(S)

Cauligi Raghavendra

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Southern California
Department of Contracts and Grants
837 West Downey Way, STO 315
Los Angeles CA 90089-1147

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington VA 22203-1714

AFRL/IFGA
525 Brooks Road
Rome NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2005-210

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Wayne Bosco/IFGA/(315) 330-3578 Wayne.Bosco@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

The objective of this project was to provide technical support to the Defense Advanced Research Projects Agency’s (DARPA) Network Modeling and Simulation (NMS) technical program. The effort acted as a facilitator for the NMS program by gathering information from program participants to build realistic network models. This effort studied traffic characterizations and mobility required to support future battlefield networking scenarios. Realistic mobility models for an urban battlefield scenario were developed. These models contained sensor nodes, ad hoc ground nodes, Unmanned Aerial Vehicles (UAV) and satellite nodes. This effort also evaluated the suitability for use in military networks, the traffic models and analysis techniques developed by other researchers in the NMS program.

14. SUBJECT TERMS Ad Hoc networks, mobility models, network simulation, network models	**15. NUMBER OF PAGES** 48
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Contents

List of Figures

1 Project Summary

Defense Advanced Research Projects Agency (DARPA)'s network modeling and simulation (NMS) program projects have developed many theoretical models for analyzing and understanding network behavior. These models are evaluated and validated using commercial network dataset or synthetically generated network traffic. Although the current intelligence and military communities believe that NMS methods and results based on commercial networks will be of great use to their planned networks, they are also concerned about some of the special characteristics of military networks. The objectives of this project are to provide technical support to the DARPA NMS program and be a facilitator for NMS participants with Department of Defense (DoD)/Intel needs. This was accomplished by studying the traffic modeling and analysis techniques developed in the DARPA NMS program and to evaluate their suitability for military networks, and to develop mobility and traffic models for future battlefield networks. Several traffic models and their use are shown for urban scenarios of future battlefield. Specifically we accomplished the following:

1. Searched for providing realistic datasets to NMS researchers from sanitized traffic data from military scenarios. This was performed through University of Southern California (USC)/Aerospace partnership with groups within National Reconnaissance Office (NRO) and wider National Security Space community.

2. We have studied the traffic characterization and the mobility in the future battlefield scenarios. This allowed us to develop realistic mobility models in urban battlefield scenarios consisting of sensor nodes, ad hoc ground nodes, Unmanned Aerial Vehicles (UAVs), satellites etc.

3. We examined all the modeling and analysis tools and techniques developed by the NMS researchers and evaluated their suitability for military network based on the traffic characteristics of military networks.

Although the results of trying to find a suitable military network traffic model/database for NMS researchers has been disappointing, we have learned a great deal that is potentially useful for the military. This has helped us identify some of the current DoD needs for network modeling, as well as their ongoing problems in creating the right datasets for themselves. One of the major things we learned from the authors of existing datasets and models was that these current datasets and models are largely limited by not reflecting the new technology and the new operational environment within networked systems. Therefore, as described below, we have first focused our study on what new types of models must be developed in order to more realistically reflect the rapidly emerging operation of networks, specifically, mobility of UAVs and Unattended Ground Sensors (UGSs) in the urban battlefield scenario. And then we have reviewed the developed models, current simulators as well as data capture and analysis tools in the context of their suitability for military network.

2 Introduction

The explosive proliferation of high performance desktop computers, high-speed networks, and the exponential growth in Internet based services have made network-centric systems touch all aspects of our life in education, industry, finance, medicine, science, government, and military. Our dependence on these systems and their information services is growing at an alarming rate in spite of the fact that our current approach to design, deploy and manage such systems and services are at best ad-hoc. Systematic design and understanding of such systems are critical to our national security and information superiority. Increasingly, many applications critically depend on the Internet infrastructure. With the exponential growth of traffic and applications over the Internet, there is growing concern about "congestion storms" and other performance anomalies. There are a number of modeling and simulation projects in the DARPA NMS program to analyze and understand the behavior of complex networked systems. For large scale networks, scalable simulations are needed to understand and predict network behavior in the presence of unanticipated events.

Currently, researchers in modeling and simulation community feel that their theoretical efforts cannot be realized into usable results of until they can test them on network traffic datasets that reflect realistically the characteristics of networks that may be special to military networks and not necessarily reflected in the current datasets based on commercial networks. Although the current intelligence and military communities believe that NMS methods and results based on commercial networks will be of great use to their planned networks, they are also concerned about some of the special characteristics of military networks. Such characteristics include the differences in priorities and loads during military events vs. peacetime or normal operation, the special problems of malicious disruption or co-opting of the network for enemy uses, and the fact that some network routes may be fully controlled, prioritized, and knowable (in other words not characterized best by statistical models).

In order to analyze network performance and evaluate unexpected problems, detailed simulations of networks are needed. DARPA NMS program has developed several simulation tools for accurately representing varying details of a network. Characterizing the traffic that flows through a network that supports users and applications is an important factor. This requires modeling and generating traffic that would be carried by a network. There is interest in characterizing traffic carried by a military network. Traffic in military networks will depend on the types of users and applications as well as whether it is peace time or war time.

In this project, we set out to systematically study traffic characteristics of military networks to look for similarities between military network traffic and commercial network traffic. The Aerospace/USC team looked into finding military network traffic datasets and explore suitable methods for sanitizing existing datasets used for NRO networking projects. However, the results of trying to find a suitable military network traffic model/database for NMS researchers has been disappointing. In our search, we learned a great deal that is potentially useful for the military.

The lack of suitable databases has been frustrating, but in the end interesting as it has helped us identify some of the current DoD needs for network modeling, as well as their ongoing problems in creating the right datasets for themselves. In the end, we refocused this study to concentrate on the current DoD needs for network modeling in urban battlefield scenario.

We systematically developed mobility and traffic generation models for UGSs in the built-up battlefield environment in Section 4. Further, by extrapolating mobility from Simulation of the Locations and Attack of Mobile Enemy Missiles (SLAMEM) logs, realistic mobility models for Organic Aerial Vehicles (OAVs) are generated in Section 5. Based on this realistic mobility models, we developed adaptive UAV placement algorithms in Section 6. We devised adaptive UAV placement algorithms that has its roots in graph theory in Section 7. We also proposed thermodynamics-based self-tuning deployment of UAVs in the urban battlefield scenario in Section 8.

We also looked at all the modeling and analysis tools developed by the researchers in the DARPA NMS program and examine their applicability to military networks. This study was based on the assumption that military networks are similar to packet-based commercial networks, but can be viewed as a Virtual Private Network. However, a military network will have its own policies for traffic handling. Since users of a military network also use commercial applications, such as e-mail, web etc., many of the traffic models developed in the NMS are program are applicable. Specific models and tools are directly applicable under certain conditions on traffic and number of flows. Our conclusion is that a majority of these models and tools can be used in military networks context.

3 Overview of current mobility models

In this section, we first, provide a brief overview of current work in the area of development of mobility models. And then we overview the current work in the area of unmanned aerial vehicle (UAV) placement problem.

3.1 An Overview of Current mobility models for Mobile Ad-Hoc Networks (MANETs)

Several mobility models have been suggested for mobile wireless network simulation. Perhaps the most widely used mobility model is random waypoint [1]. This model restricts the movement of the mobile nodes to a rectangle. Each node picks a destination within the rectangle along with a speed. The node travels to the destination at the selected speed. Upon reaching the destination, the node selects and waits for a uniformly distributed pause time. After waiting, the node picks another destination and another speed, continuing the process. The parameters of this model are the minimum and maximum speed and the maximum pause time. It is not uncommon that the speeds range from 0 to 20meters/sec.

Besides random waypoint, there are several other models that guide nodes along random paths through a rectangle. These include Random Walk Mobility Model [2] that selects directions and speeds at random and Gauss-Markov Mobility Model [3].

Another class of models is those that restrict nodes to a graph. These can be subdivided into graph-based random walk model and graph-based random waypoint model. In a graph-based random walk model, the nodes move from vertex to vertex, selecting the next destination at random from the neighboring vertices. A graph-based random waypoint model selects a destination at random from all vertices or a set of vertices. The node then moves along the graph to the destination vertex. In most cases, the speed of the node is selected at random and the pauses may occur when the destination is reached. The Manhattan mobility model is a graph-based random walk [4] where the graph is a 2-D lattice. The City Section model is similar to Manhattan, but uses random waypoint and restricts the speed of the nodes to resemble model traffic moving along city streets. Another graph-based random waypoint model is presented in [5], where the graph was defined by a Voronoi diagram of obstacles. This graph was further extended to include the vertices of the center point of the obstructions and arcs that emanate from the center of the obstruction to the arcs of the Voronoi diagram.

Other classes of model are group mobility [6] and Scenario based models [7]. However, as mentioned in [7], these scenarios are not intended to be realistic.

There are detailed transportation simulators such as TRANSIM and CORSIM. Such simulators are strictly focused on estimating automobile congestion, with special focus on highway congestion. These models may be large scale in that they can span entire cities.

3.2 Overview of current UAV placement work

Recently, there has been a number of works that have addressed different aspects of UAV placement when enough number of UAVs are placed in the network to provide direct coverage for all nodes as a backbone regardless of the node-to-node communication capability of the network. In [8] and [9], a novel architecture with ground nodes, UAVs and also unmanned ground vehicles is introduced and the performance of the overall network is evaluated by simulation studies. However, the UAV placement is with respect to the full coverage of the nodes which resembles the classic p-center problem and some other variations of it. [10] mainly deals with the implications of the UAVs on the routing protocol and its performance. The main source of related work on similar problems is the literature about the Facility Location problem and specifically the p-center problem. It is shown in [11] and [12] that the p-center problem is NP-hard (nondeterministic polynomial hard). Exact solutions are provided by [13], [14], [15], [16] all address the Euclidean distance p-center problem but involve rather inefficient schemes that do not scale to larger problems. There are also several heuristic methods for the p-center problem that are based on the assumption that the single facility location problem can be efficiently solved. All of those methods quickly find locally optimal solutions though. In [17] paper, the authors adapt the deterministic annealing clustering algorithm to their formulation and use that to find the minimum number of UAVs required to provide connectivity and their locations. In [18], the UAV placement and navigation strategies with the end goal of improving networking connectivity is examined. The authors propose to use local flocking rules that aerial living beings like birds and insects follows, to meet their goals.

4 Mobility and traffic models for Unattended Ground Sensors (UGSs)

The objective is to develop realistic network mobility and traffic models for Unattended Ground Sensors (UGSs) in urbanized settings. We take the approach described in Figure 4.1. Given the map of the urban area, we generate mobility and traffic models based on the characteristics of the UGSs and the targets. These simulations generate QualNet [19] config files as an output. QualNet simulations are ran based on these config files and the results are generate. At this stage there is no feedback loop. Meaning that the results from the QualNet simulator are not fed back to modify mobility and traffic generation models. However, this could be done to achieve desired performance.We require the building map and specification of sensors and the target/targets. The building map could be generated from Geographic Information Systems (GIS)/Geography Markup Language (GML) [20] shape files. The target moves according to constrained random waypoint model. Meaning that, the target moves randomly but it doesn't hit any buildings. For UGS, two settings were considered. 1) The UGSs are stationary. 2) The UGSs are moving. For a particular scenario shown in Figure 4.2, the sensors were stationary. Sensors will generate traffic when the target is in the range and there is a line of sight between the sensors and the target. Sensors follow ON/OFF traffic model. When the sensors are ON, they generate traffic at a constant rate depending on a type and specifications. At the end of a matlab simulation, we get two files in the form of QualNet input files: 1) mobility file (.mobility) 2) application files (.app). These files are fed into QualNet and the resulting statistics are collected.

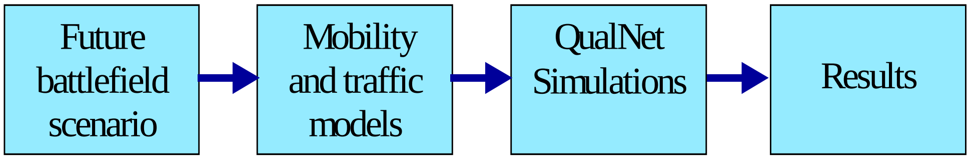

Figure 4.1: Block diagram

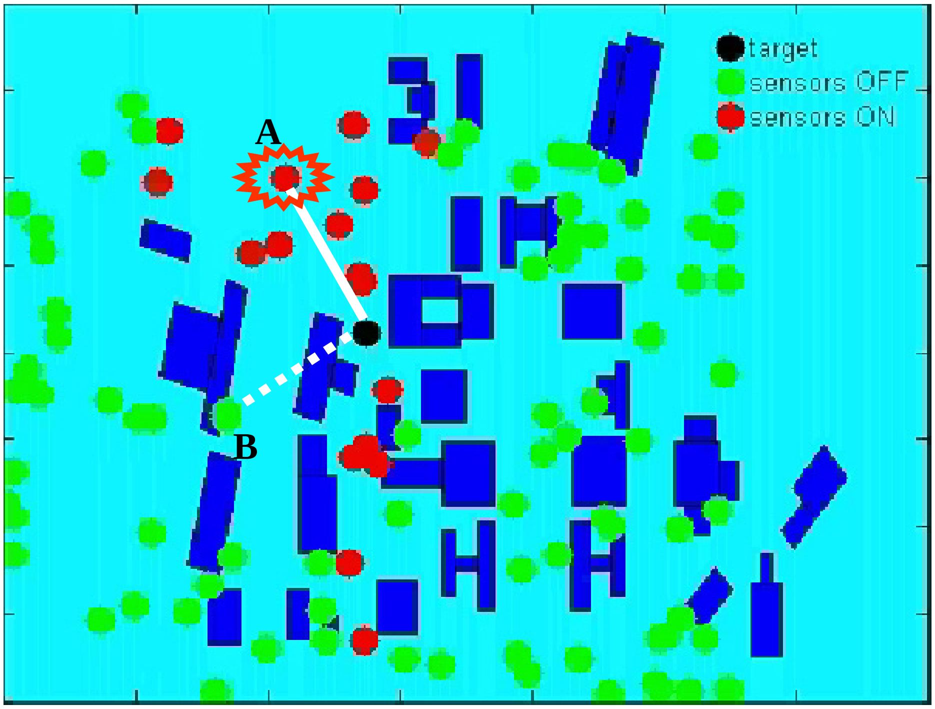

Figure 4.2: As the target moves, the sensors generate traffic if the target is in its range and there is a line of sight between the sensor and the target. The blue blocks show the 2-D view of buildings.

5 Advanced mobility models for Organic Aerial Vehicles (OAVs)

In this section we develop realistic mobility models for Organic Aerial Vehicles (OAVs) in urban environment. We take the following approach to model the mobility of OAVs. Since, we have an access to the Simulation of the Locations and Attack of Mobile Enemy Missiles (SLAMEM) logs, we first extract out the mobility information of the various types of OAVs from the logs. Then we model this mobility of the OAVs based on the trace information.

The Unmanned Aerial Vehicles (UAVs) fly at a very high altitude (~20000m) while the Middle Altitude Unmanned Aerial Vehicles (MAUAVs) fly at a medium altitude (~5000m). The Organic Aerial Vehicles (OAVs) can fly at a very low altitude (0-1000m). Various types of UAVs, MAUAVs, and OAVs are shown in the figure 5.1.

Mobility models for UAVs are of less interest as these vehicles tend to fly at higher altitude in circles to cover the entire space. The mobility traces for UAVs from SLAMEM logs have shown that. Hence, we concentrate on finding the mobility models of the OAVs. These OAVs can fly at a very low altitude. In the urban setting this becomes even more of an interest, as they can fly at a building height. Since there is always a variation in the building heights, OAV's trajectory can be affected by that.

The following approach is taken to derive the mobility models of OAVs. First, we use the SLAMEM logs and extract out the real mobility of the OAVs. From the logs we see that the OAVs can be in one of the following stages.

1. PATROL
2. RETURNING_TO_BASE_FOR_SUPPLIES
3. NOT_READY_AT_BASE
4. TAKEOFF
5. READY_AT_BASE
6. CANDIDATE_AT_BASE
7. NOT_AT_5

As an OAV spends most of the time patrolling, we only focus on OAV's trajectory when it is patrolling. The following figure shows the trajectory of an OAV extracted from the SLAMEM logs. Figure 5.3 shows one example of the 3-D view of an OAV's trajectory. Figure 5.4 shows the 2-D view of the same OAV's trajectory. It can be seen from the figure 5.4 that the 2-D OAV's trajectory is made up of many polygons-like shapes. We call these closed shapes primitives. A primitive is defined as a closed loop that an OAV traverses at least twice.

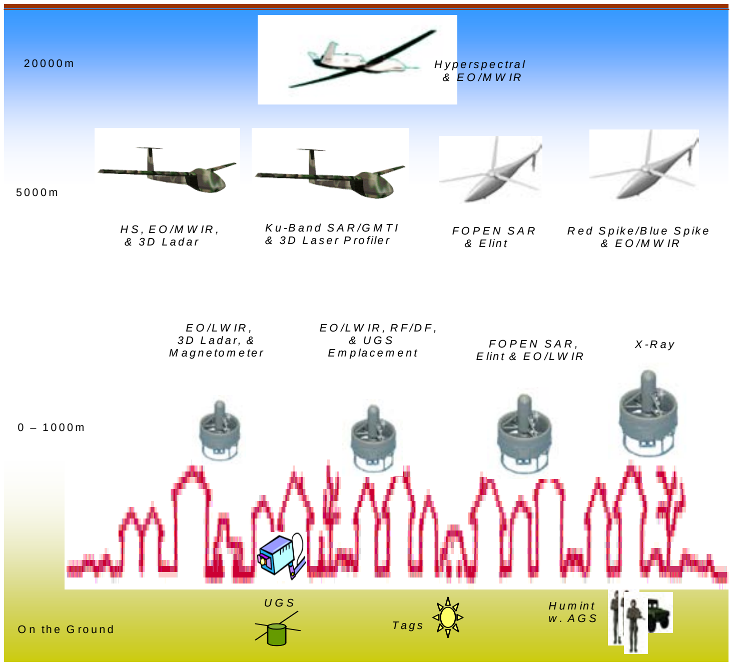
20000m
Hyperspectral & EO/MWIR
5000m
HS, EO/MWIR, & 3D Ladar
Ku-Band SAR/GMTI & 3D Laser Profiler
FOPEN SAR & Elint
Red Spike/Blue Spike & EO/MWIR
EO/LWIR, 3D Ladar, & Magnetometer
EO/LWIR, RF/DF, & UGS Emplacement
FOPEN SAR, Elint & EO/LWIR
X-Ray
0 – 1000m
UGS
Tags
Humint w. AGS
On the Ground

Figure 5.1: Classifications of the UAVs, MAUAVs, OAVs, UGSs, and Attended Ground Sensors (AGSs)

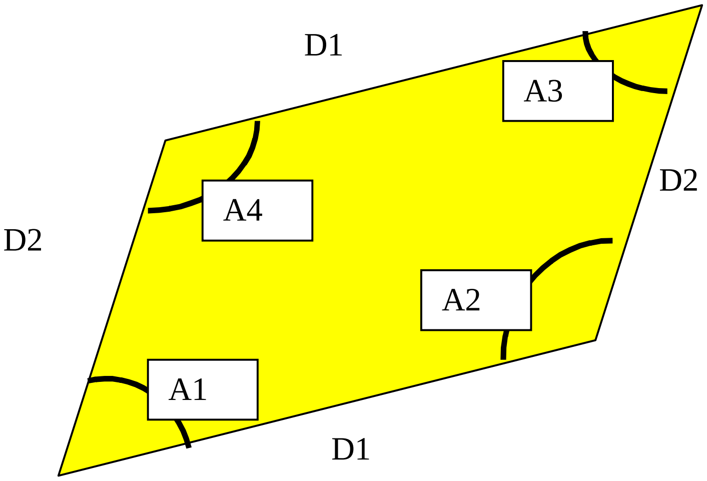

Figure 5.2: A four-sided polygon (primitive)

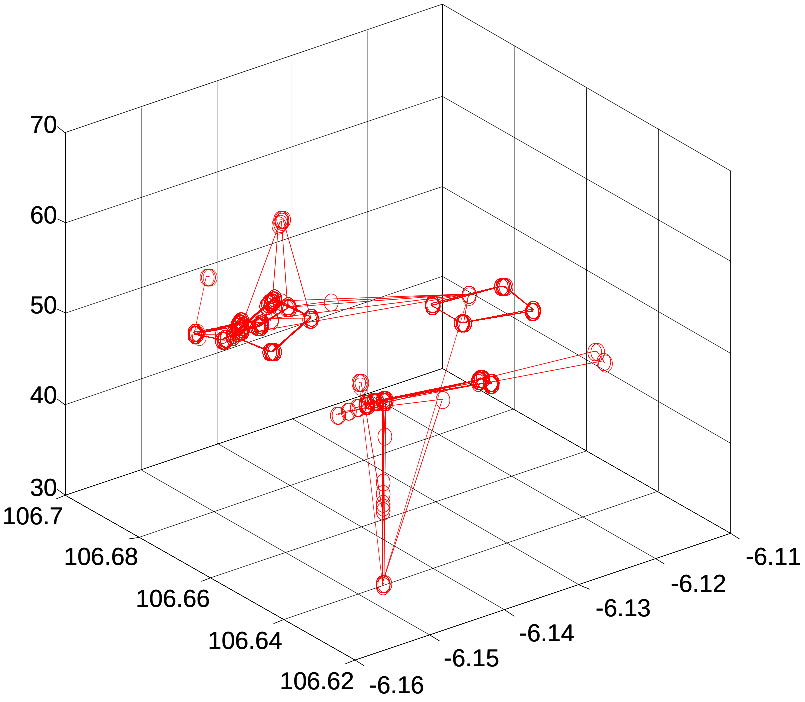

Figure 5.3: A 3-D view of an OAV's trajectory

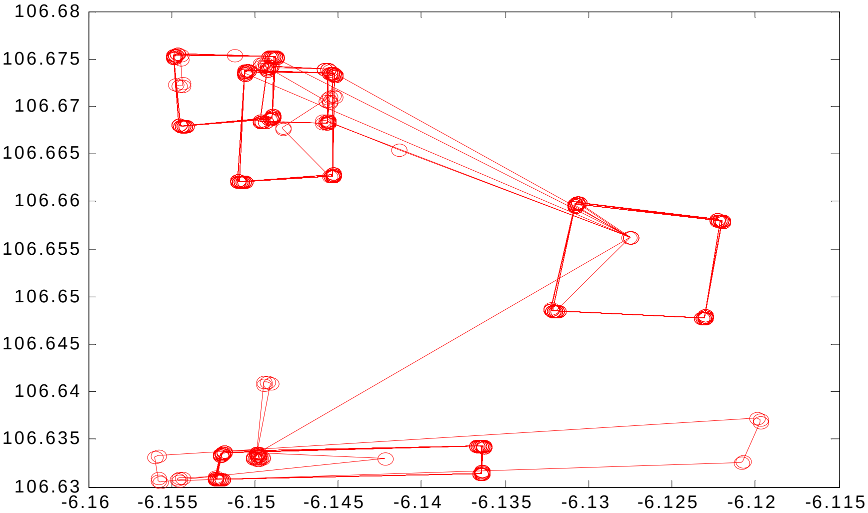

Figure 5.4: A 2-D view of an OAV's trajectory

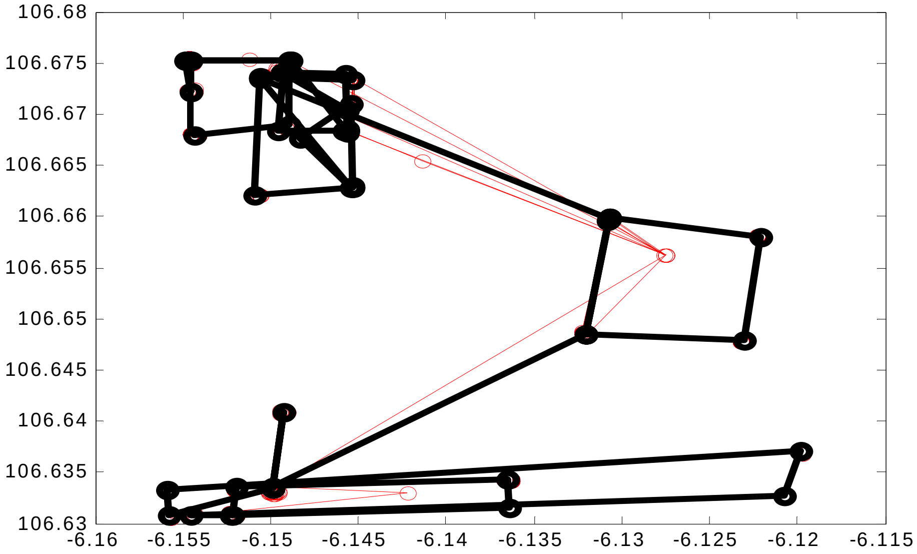

Figure 5.5: A 2-D view of an OAV's trajectory. The black lines show the primitives found by primitive-finder.

To develop a mobility model, first, all the primitives and their parameters must be found.

Each primitive can be parameterized by sequences of distances and angles. For example, the Figure 5.2 shows a four-sided polygon. We need to know the distances D1, D2 and the interior angles A1, A2, and A3 to be able to draw this primitive. However, there is a possibility that the polygons are irregular. To include those cases, we parameterize an n-sided primitive by the distance of (n-1) sides and (n-1) interior angles.

We developed an algorithm that determines all the possible primitives from a given OAV's trajectory. This algorithm uses the idea of grouping the close points together and finding the loops between sets of close points. Figure 5.5 shows the primitives found by using our primitive-finder. The red solid line shows the actual OAV trajectory while the black solid line shows the primitives found by the primitive-founder. We can see that primitive-finder finds all of the primitives from that OAV's trajectory. Figure 5.6 shows even more complicated example of an OAV's trajectory. The red solid line shows the OAV's original trajectory and the black solid line shows the found primitives.

After processing all the SLAMEM logs for OAVs, we get all the possible primitives by using our primitive-finder. We first find all the distributions from these primitive that are necessary to model the mobility of an OAV. To completely define the primitives and their parameters the following distributions are computed.

1. The distribution of the number of corners in each primitive

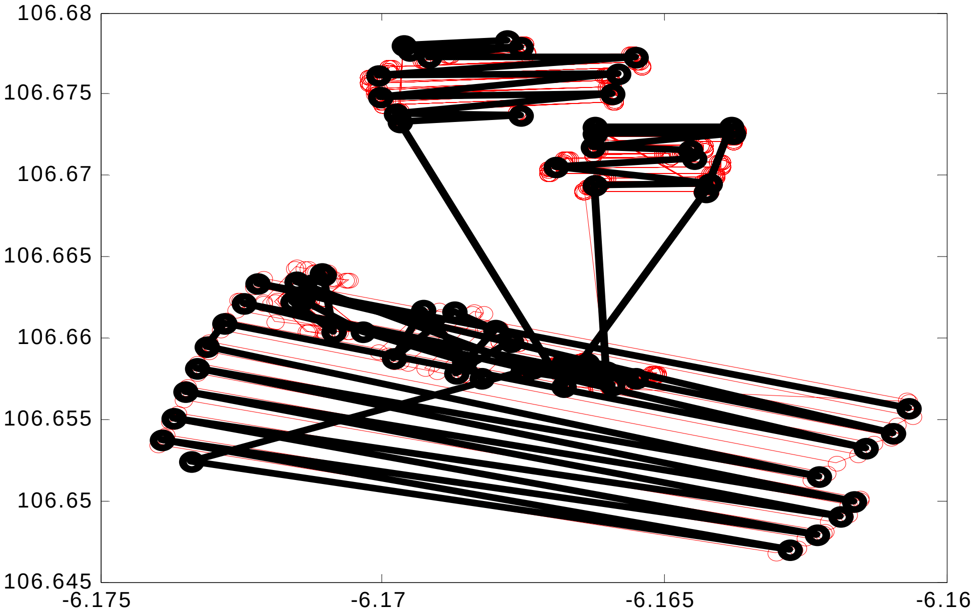

Figure 5.6: A 2-D view of a more complicated OAV's trajectory. The black lines show the primitives found by primitive-finder.

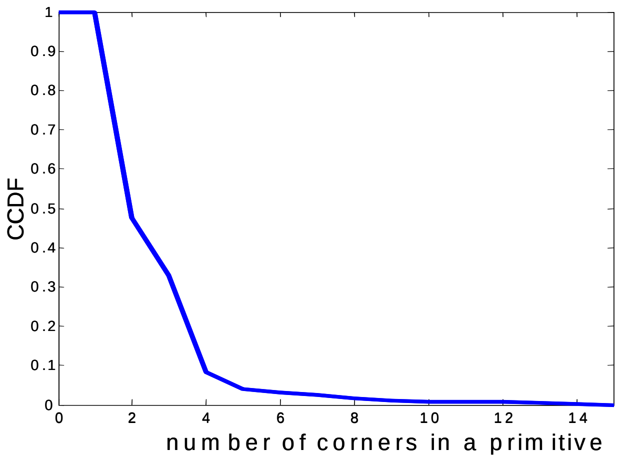

Figure 5.7: Complementary Cummulative Distribution Function (CCDF) of the number of corners in a primitive

2. The distribution of the time spent in each primitive

3. The distribution of the distances between two vertices in each primitive conditioned on number of corners in each primitive.

4. The distribution of the interior angle between two edges in each primitive conditioned on number of corners in each primitive

5.1 The distribution of the number of corners in a primitive

Figure 5.7 shows the complementary cumulative distribution function (CCDF) of the number of corners in a primitive. We can see that 30% of the times the OAV has only one corner in a primitive. This would mean that the OAV is hovering around a particular point. Observe that 50% of the time a primitive has more than 2 corners. Also note that probability of finding a 10-or-more corner primitive is less than 1 %. We find that the maximum number of corners a primitive can have is 15. However, probability of getting 15 corner-primitive is very low.

5.2 The distribution of the time spent in a primitive

Figure 5.8 shows the CCDF of the time spent in each primitive. Figure 5.8 shows that 70% of the times the time spent in a primitive is larger than 120 sec. Meaning that, 30% of the times the time spent in a primitive is smaller than 2 min. It can be seen that 50% of the times, the time spent is a primitive is larger than 500 sec. The part of the distribution where the time

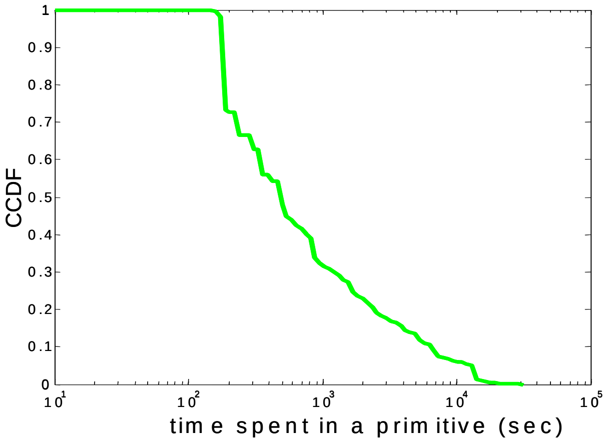

Figure 5.8: CCDF of time spent in a primitive

spent in a primitive is large than 2 min. can be modeled as Pareto distribution. Also observe that with a very small probability (0.005), we have OAVs that spent about 20000 sec (5.55 hours) in a single primitive.

5.3 The distribution of the distances between two vertices in a primitive

Figures 5.9 and 5.10 show the CCDF of the distance between two vertices in a primitive. This distribution depends on the number of corners in a primitive. Hence, we compute the CCDF for the distance between vertices conditioned on the number of corners in a primitive. Figure 5.9 shows this CCDF conditioned on four-corner primitives while the Figure 5.10 shows this CCDF conditioned on five-corner primitives. Notice that the CCDF of the distances is steeper when conditioned on 5-corner primitive as compared to 4-corner primitive. This means that the distance between the vertices tend to be larger for a 5-corner primitive as compared to a four-corner primitive. Make a note that these distances are in degrees of arc length. 1 degree of arc length = 60 nautical miles = 60 ×1852 meters =111.12 km. Hence, very little difference in degree of arc length accounts for a huge difference in meters.

5.4 The distribution of the interior angle between two edges in a primitive

Figures 5.11 and 5.12 show the CCDF of the interior angles of a primitive. Interior angles of a primitive depends on the number of corners in that primitive. Hence, we compute the CCDF of

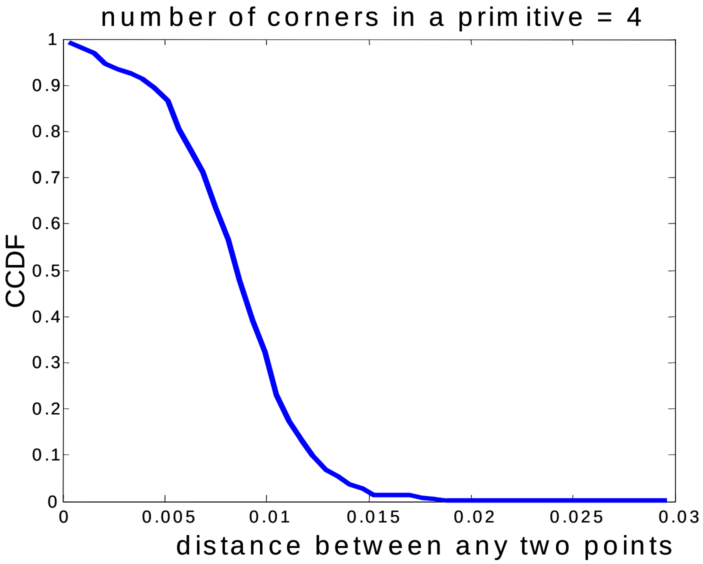

Figure 5.9: CCDF of the distance between two vertices in a primitive conditioned on number of corners in a primitive = 4.

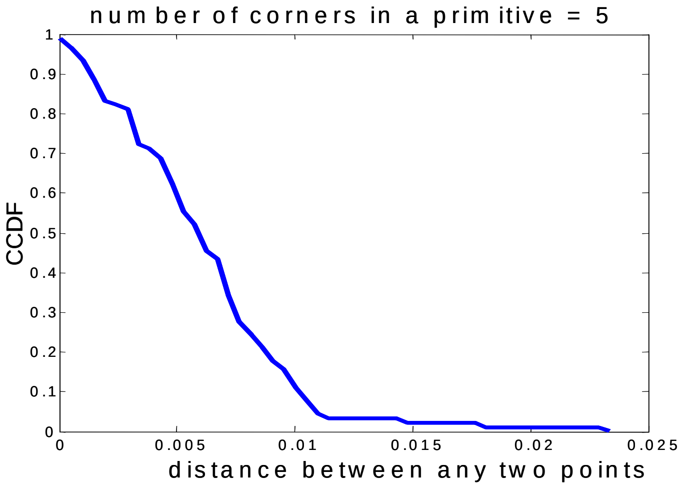

Figure 5.10: CCDF of the distance between two vertices in a primitive conditioned on number of corners in a primitive = 5.

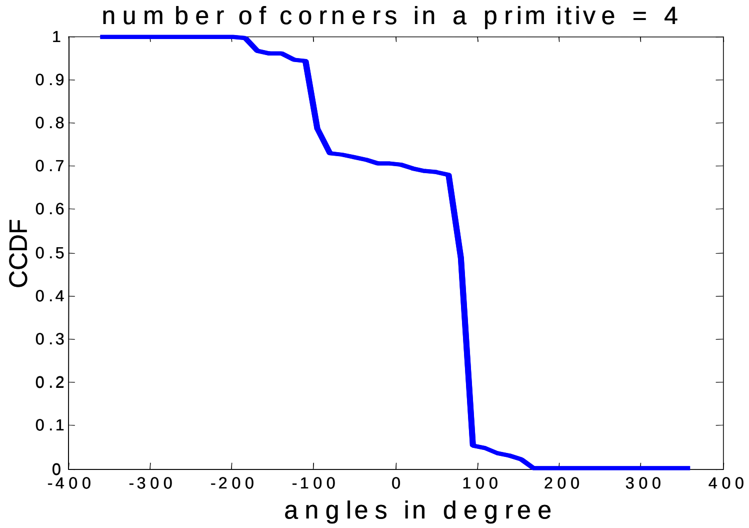

Figure 5.11: CCDF of the interior angles in a primitive conditioned on number of corners in a primitive= 4.

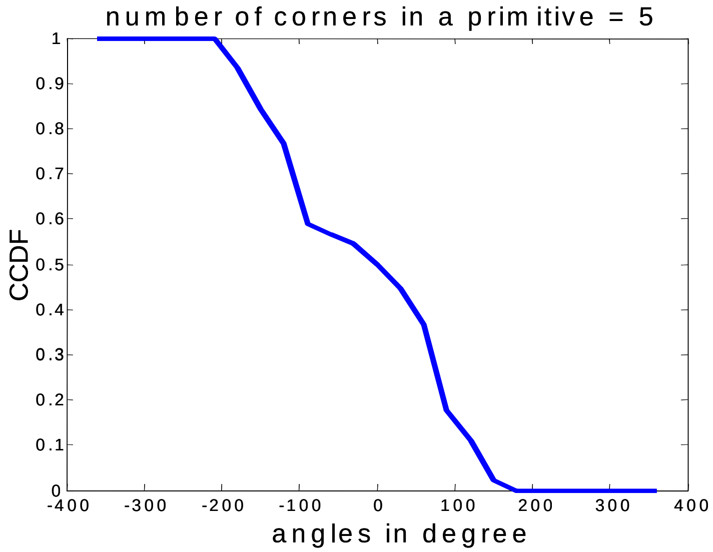

Figure 5.12: CCDF of the interior angles in a primitive conditioned on number of corners in a primitive = 5.

the interior angles conditioned on the number of corners in a primitive. Figure 5.11 shows this CCDF conditioned on a four-corner primitive and the Figure 5.12 shows this CCDF conditioned on a five-corner primitive.

Figure 5.11 shows that the 90% of the times the interior angles are bigger than –110 degrees. Furthermore, the interior angles are positive 70% of the times. However, the interior angles are above 90 degrees only 0.5 % of the times. Figure 5.12 shows the CCDF of the interior angles for five-corner primitives. In this case, 90 % of the times, the interior angles are bigger than –180 degrees and half the times the angles are positive.

6 UAV placement in an urbanized setting

To date, most ad hoc networking research focuses on networks operating in an open area with no hills, no trees, no buildings, and arguably no ground. However, the military has some interest in such an environment (e.g. desert or airborne MANETs). There is also interest in battle scenarios in urban areas (e.g., Bagdad). Hence, today, it has become important to know the optimal placement of UAVs in the urban battlefield scenario. In this section, we study the number and the placement of the UAVs over the ground nodes in the urban battlefield scenario with the end goal of improving coverage in the ad hoc network.

The coverage problem is straightforward in the case of free-space terrain as compared to a build-up urbanized environment. In the urbanized setting, we have buildings that can *obstruct* the view. If we think of an UAV as a source of light then the effects of the buildings can be seen as a shadowing effect. Meaning, UAVs can not see a target if it is in the shadow of the building. This problem does not occur if we have a free-space environment. Another problem can arise is the heterogeneous nature of the terrain. The density of the buildings can be different in various parts of the terrain. It becomes difficult to find the UAV density that can provide uniform coverage for the entire space. This problem does not arise in the free-space setting, as this environment is homogeneous.

In the urban area, the coverage on the ground can be affected by a number of following factors:

1. The nature of the terrain.
2. The sensing range of the UAV sensors
3. The density of the UAVs
4. The mobility of the UAVs
5. The mobility of the ground nodes

We have developed various algorithms that finds the number and the placement of the UAVs over the ground nodes in the urban battlefield scenario.

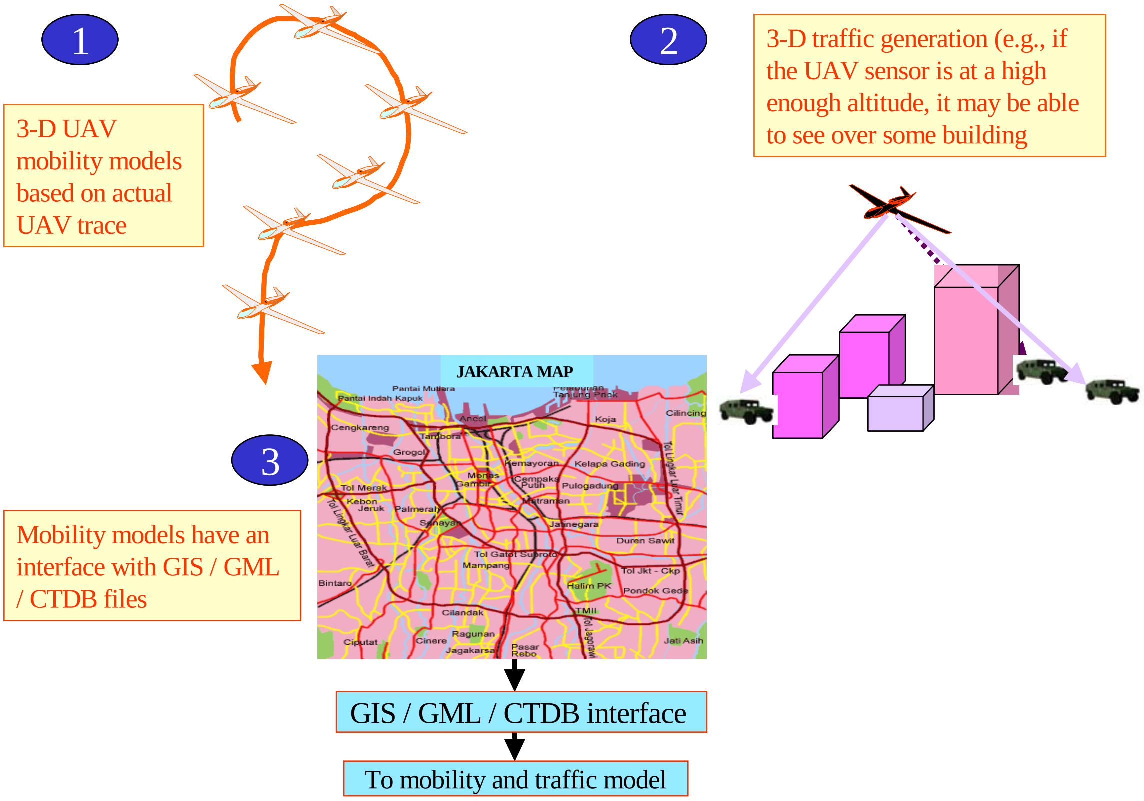

Figure 6.1: A pictorial view of the algorithm

6.1 Algorithm 1: UAV coverage algorithm based on realistic mobility models

This algorithm proceeds in the following steps. The figure 6.1 shows the pictorial view of this approach.

1. Find the realistic mobility model of UAV/OAV.

2. Incorporate GIS/GML/CTDB ([20], [21], [22]) shape files to create the building map.

3. Find the UAV/OAV coverage given the design specifications (sensor transmission range, number of UAV/OAV, etc.)

6.2 Description of the algorithm

This algorithm uses the mobility models of OAVs we derived in the section 5. We obtained several distributions in the section 5. We use the distributions of the number of corners, distance conditioned on the number of corners, interior angles conditioned on the number of corners to derive the mobility of the OAVs.

The coverage problem can be formulated in the following manner. The whole space is divided into two parts: The ground space and the aerial space. Ideally, we would like to cover the entire ground space. Instead we divide the whole ground space into many lattice points. And the problem boils down to the coverage of the ground lattice points.

Consider the n ground lattice points $L = \{L_1, L_2, \ldots, L_n\}$ and the m aerial lattice points (OAVs) $O = \{O_1, O_2, \ldots, O_m\}$. We want to find the coverage of the ground lattice points given the aerial nodes. The performance of this algorithm can be defined by two metrics. 1) The fraction of the time (FT) each ground lattice point is covered by any of the aerial lattice point 2) the mean time (MT) any lattice point is ON, meaning that, the mean time a particular lattice point is tracked by one of the OAVs. Notice that these two metrics are different. The first metric describes the fraction of time that a ground lattice point is viewed by any of the OAVs. The FT = 0.5 will mean that 50% of the times the lattice point was seen. If only this metric was considered, we will not be able interpret anything about the actual time between viewing. The FT = 0.5 can mean that the lattice point was seen every other second and it can also mean that the lattice point was seen every other day. Hence, perhaps more relevant metric is the second one. This metric gives an idea of tracking time duration of a ground lattice point. If we are given FT = 0.5 and MT = 10 sec, this can be interpreted as, the ground lattice point is seen 50 % of the times. And whenever that lattice point is viewed it is on an average viewed for 10 sec. Hence, these two metrics together define the performance of the algorithm. In the ideal setting, the ground lattice point should be seen the entire time (FT = 1). We can improve the coverage by adding more OAVs, however, there is a trade-off between the cost of additional OAVs and the performance.

Number of ground lattice points (n) and the number of aerial lattice points (m) is known. To achieve the goal (the target performance metrics), we first find the mobility of the OAVs based on the mobility model described in the section 5. In this setting, the ground targets are stationary. However, it should be noted that in the urbanized setting, we have to consider the *obstruction* caused by the buildings. The ground target could have a line-of-sight with an OAV, and could be in the sensing range of that OAV. However, we have to make sure that this target is not in the shadow of the building. While keeping this in mind, we compute both the metrics. The figures 6.2 and 6.4 show the metrics.

The following describes the step-by-step algorithm:

1. Given n and m.

2. For each OAV O_i pick the number of corners (N) in a primitive from the distribution of the number of corners shown in section 5.

3. For a given N for the O_i, pick distances between the vertices in a primitive from the distribution of the distance given N.

4. For a given N for the O_i, pick interior angles of a primitive from the distribution of the

angles given N.

5. After finding mobility of each O_i , based on the sensing range of each O_i and the shadowing effect, find the two metrics: FT and MT for all the ground lattice points.

In this experiment setup, we fix $m = 15$ and $n = 135$. Figure 6.2 shows the fraction of time the ground lattice point was seen. The x-axis and y-axis are the x-coordinate and y–coordinate of the ground lattice points. The z-axis shows FT for each ground lattice point. Figure 6.2 illustrates that the overall mean coverage is 0.6 while the maximum coverage at several lattice points is 1. However, the minimum coverage is 0. This shows the variability in the coverage and shows that the coverage is heterogeneous. Notice that, the mean fractional coverage and the worst-case coverage are quite different. Thus, using only mean fractional coverage as a metric is not sufficient.

Figure 6.3 demonstrates the top view of the figure 6.2. This figure clearly shows the effect of the buildings. For instance, for x-coordinate 100 and y-coordinate 4, we see that coverage is zero. As we move further from this ground lattice point in either of the directions the coverage slowly increases. This shows that there is a building in that area. Since the buildings cast shadows, the coverage starts to increase as we move away from the shadowing effect. This effect can be seen throughout the entire plot. In the open area the coverage is better.

Figure 6.4 shows the MT for each ground lattice point. The x-axis shows the index of the lattice point while the y-axis shows the corresponding mean on time. This plot shows the variation in the mean on time each lattice point is tracked by any OAV. Typically, each lattice is point is tracked for 11.25 sec with a standard deviation of 6.5 sec. However, we can see that there are some lattice points that have MT of 1 sec. These lattice points certainly are affected by the buildings shadows.

Overall, the ground coverage in urbanized setting is heterogeneous and so is the performance this algorithm.

6.3 Algorithm 2: Adaptive UAV placement algorithm

The results from the algorithm 1 clearly showed the effect of heterogenous urban environment on the coverage. In this section, we develop an adaptive UAV placement algorithm that takes this effect of heterogeneity into account. This method is very similar to adaptive numerical integration. The basic idea behind the algorithm is as follows: First take the entire aerial space and compute performance metrics (FT and MT). If the metrics do not meet the desired goal then sub-divide this aerial space into four equal sized regions. For each region compute the performance metrics again. For a particular region, if the performance goal is not met then sub-divide that region into four equi-sized regions. Repeat this recursive algorithm until the desired performance is achieved.

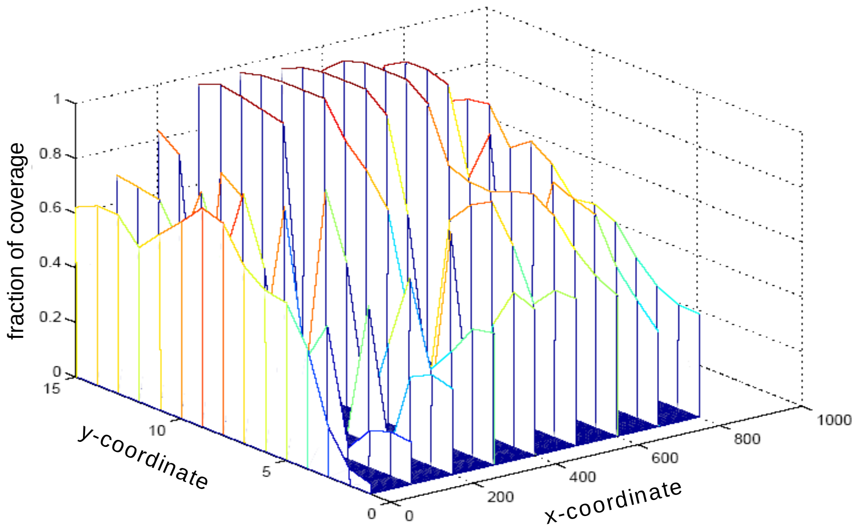

Figure 6.2: The figure shows the fraction of time each ground lattice point is covered.

In a specific region, UAV patrols in the systematic zigzag fashion. Figure 6.5 illustrates the trajectory of a patrolling UAV in a particular region.

Figures 6.6 and 6.7 show 3-D and 2-D view of the resulting UAV placement from this algorithm respectively. There is a clear sub-division of the aerial space. It can be seen that where there are more buildings there is a fine partition in the space and where there are no buildings there is a coarse partition. As discussed before, in the built-up area we require more UAVs to achieve the desired coverage as compared to open area. The region that have coarser partitions requires less number of UAVs as compared to the finer partitions. This is demonstrated in the figures 6.6 and 6.7.

In summary, this algorithm provides a better control over the coverage. The area where there is less coverage is detected and more UAVs are added in that area to meet the desired objective. Using this algorithm, the worst case coverage can be achieved in a reasonable way. However, this algorithm is computationally intensive because it takes the iterative approach.

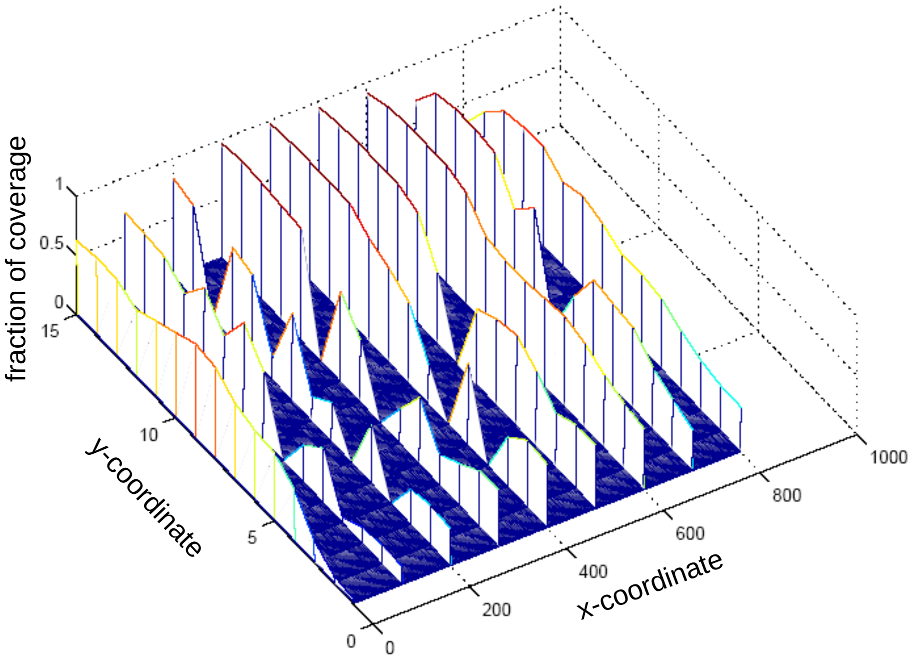

Figure 6.3: The figure shows the fraction of time each ground lattice point is covered from a different angle. The coverage is zero where there are buidings.

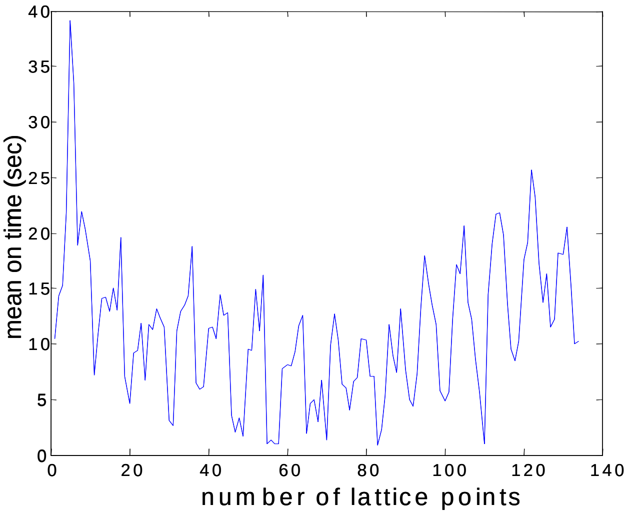

Figure 6.4: Mean on time vs lattice points.

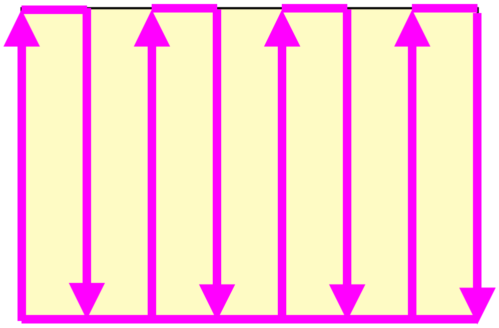

Figure 6.5: Trajectory of an UAV in a particular region.

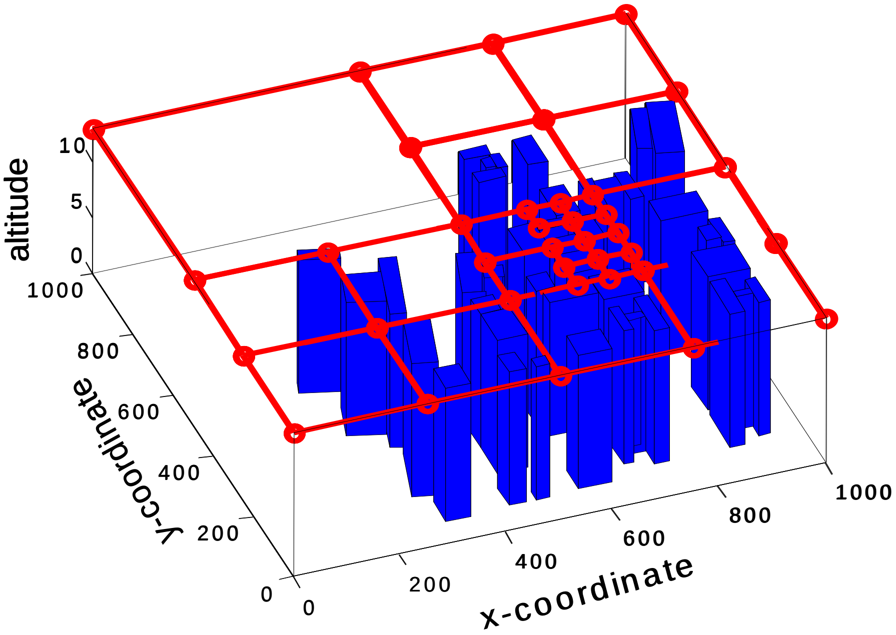

Figure 6.6: A 3-D view of the adaptive UAV placement algorithm.

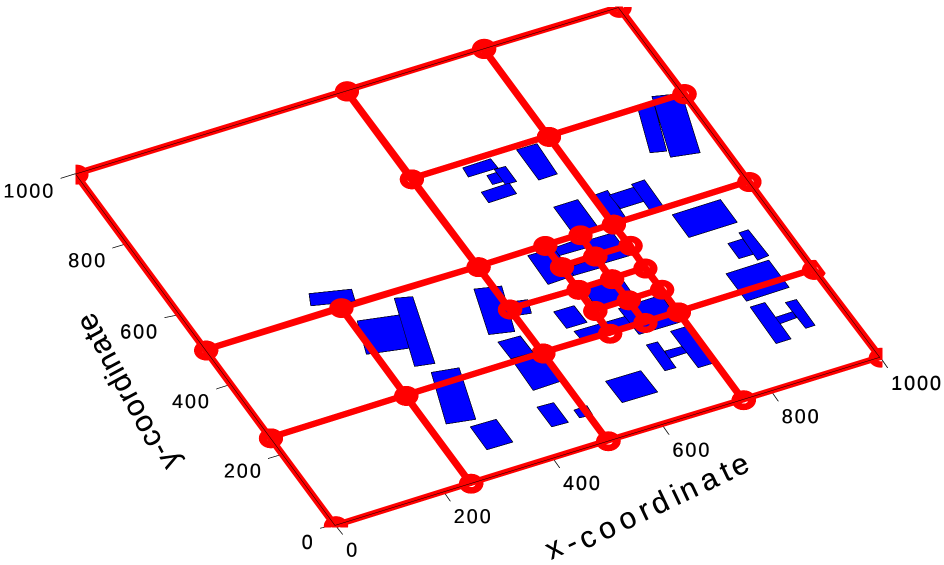

Figure 6.7: A 2-D view of the adaptive UAV placement algorithm

7 UAV placement in urbanized setting from graph theory point of view

In this section we develop the algorithms for UAV placement with the end goal of finding an optimal number of UAVs to cover the entire ground space. Before we discuss our approach, we review some definitions from the graph theory.

Definition 7.1 *Neighbors of any vertex in a graph are defined as the neighbors of a vertex of all the vertices which are connected to that vertex by a single edge.*

Definition 7.2 *A dominating set for a graph is a set of vertices whose neighbors, along with themselves, constitute all the vertices in the graph.*

Definition 7.3 *A minimum dominating set for a graph is a* minimum *set of vertices whose neighbors, along with themselves, constitute all the vertices in the graph.*

Definition 7.4 *A bipartite graph is a set of graph whose vertices are decomposed into two disjoint sets such that no two graph vertices within the same set are adjacent.*

We divide the entire graph (space) into two parts: the aerial space and the ground space. Since the aerial space and the ground space are two disjoint sets of a graph, this can be viewed as a bipartite graph. Each part of the graph has its vertices: the aerial nodes and the ground nodes. Every ground node is a neighbor with at least one of the aerial nodes. The goal is to find the minimum set of aerial nodes such that each ground node is neighboring with at least one of the aerial nodes. This is similar to finding the minimum dominating set in one half of the graph because this graph is a bipartite graph. However, problem of finding minimum dominating set is a NP-complete. Hence, this problem is also NP-complete. Therefore, the optimal solution to this problem is not possible. Instead, we pick a heuristic based greedy algorithm. We study two variations of greedy algorithm.

7.1 Greedy algorithm I

In a bipartite graph, we have a set of aerial vertices and a set of ground vertices. Each aerial vertex has a list of its ground neighbors A ground vertex is called a neighbor with an aerial vertex if the ground vertex is covered by the aerial vertex. For each aerial lattice point, we find the list of the ground lattice points it can see based on the algorithm described earlier. This algorithm used the notion of sensing range and the shadowing effect to compute the coverage.

At the end of this process, we have a set of ground nodes each UAV can observe. The goal is to find a minimum set of aerial vertices such that it covers all the ground nodes. This algorithm is greedy in the sense that the aerial vertex that covers the most of the ground vertices is picked first.

Algorithm 7.5 *A bipartite graph B has two sets of vertices: the aerial vertices set* $A = \{a_{1,}a_{2,}..., a_n\}$ *and the ground vertices set* $G = \{g_{1,}g_2, ..., g_n\}$*.For each* a_i *find a subset* S_i *of* G *such that every ground vertex in set* S_i *is a neighbor with the aerial vertex* a_i*.*

```
Begin loop:
    Pick the aerial vertex $a_i$ that covers the maximum number of ground vertices. Say, the aerial vertex $a$ has the maximum number of neighbors and $S = \{g_1, g_3, g_7\}$.
    $U = \{a\}$;
    $A = A - \{a\}$;
    $G = G - S$;
    For each $a_i$,redefine $S_i$;
    Repeat this until $G = \{\}$,meaning all the ground nodes are covered. The resulting $U$ is the set of aerial nodes that can cover all the ground nodes.
end loop
```

7.2 Greedy algorithm II

This algorithm has the same goal as the greedy algorithm I. However, this algorithm differs from the greedy algorithm I in how the aerial nodes are picked. Instead of picking the aerial node that covers the maximum number of the ground nodes, we pick the aerial node that covers the maximum number of least covered ground nodes. The rationale behind this is if we pick the aerial nodes that have high degrees, those nodes tend to cover the ground nodes that are easy to cover. To cover the difficult part of the ground space (eg. where the buildings are), we might have to add more aerial nodes. If we try to pick the aerial nodes that cover the rarely covered ground nodes, then most likely we will cover easy places as well. The vice versa might not be true.

Algorithm 7.6 *A bipartite graph B has two sets of vertices: the aerial vertices set* $A = \{a_{1,}a_{2,}..., a_n\}$ *and the ground vertices set* $G = \{g_{1,}g_2, ..., g_n\}$*.For each* a_i *find a subset* S_i *of* G *such that every ground vertex in set* S_i *is a neighbor with the aerial vertex* a_i*.*

```
Begin loop:
    Pick the aerial vertex $a_i$ that has the maximum number of the least covered ground vertices. Say, the the aerial vertex $a$ has the maximum number of neighbors and $S = \{g_1, g_3, g_7\}$.
    $U = \{a\}$;
    $A = A - \{a\}$;
```

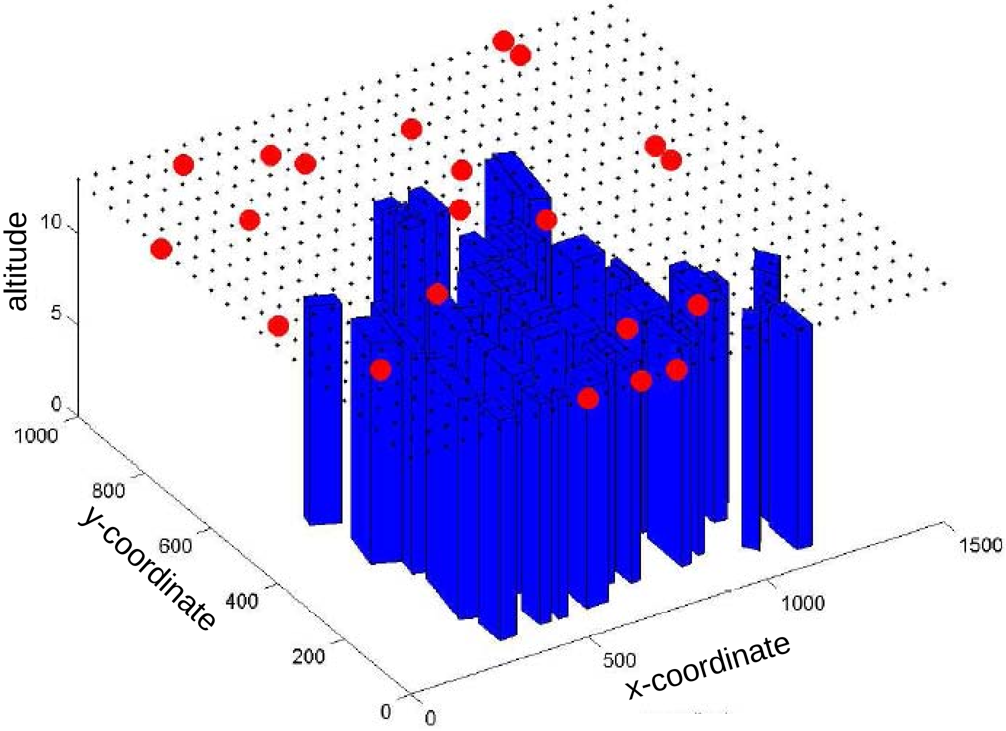

Figure 7.1: UAV placement based on greedy algorithm I.

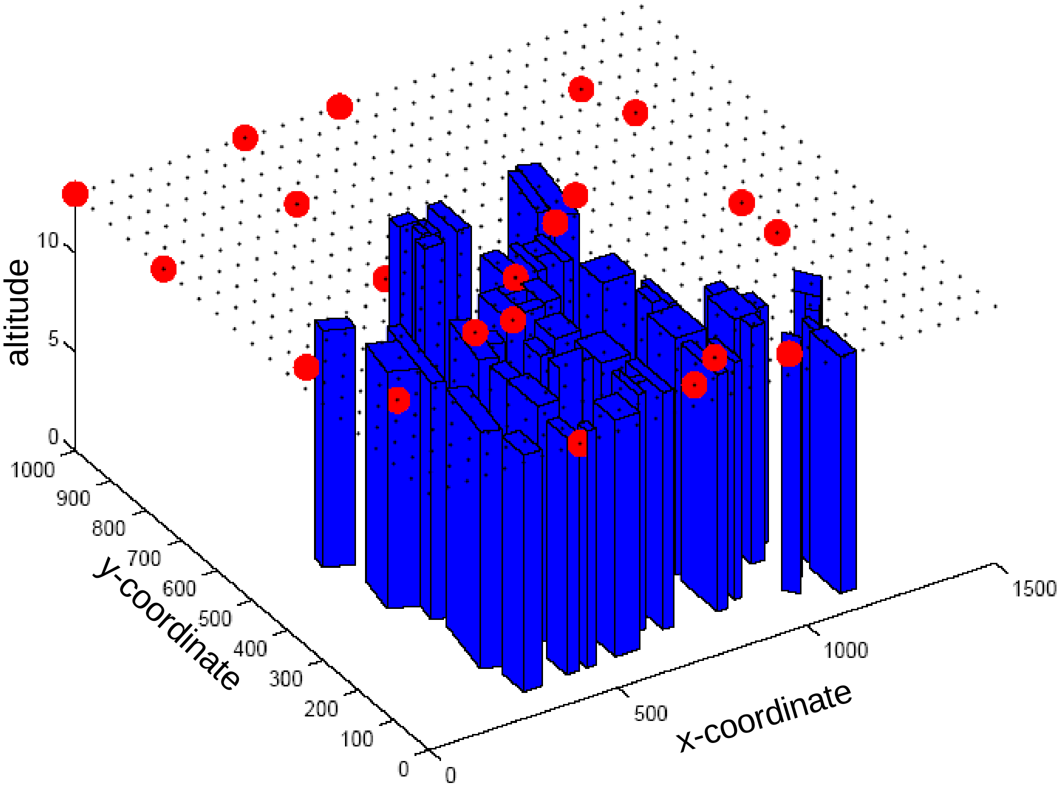

Figure 7.2: UAV placement based on greedy algorithm II

$G = G - S$;

For each a_i,redefine S_i;

Repeat this until $G = \{\}$,meaning all the ground nodes are covered. The resulting U is the set of aerial nodes that can cover all the ground nodes.

end loop

7.3 Performance evaluation

Figure 7.1 illustrates the resulting placement of UAVs from greedy algorithm I and the figure 7.2 shows the resulting UAV placement from greedy algorithm II. In both the figures, the small black dots show the set of aerial nodes. For the example shown here, we considered a set of 50 aerial nodes. The large red dots show the selected aerial nodes based on the algorithm. For both the algorithms, the placement of the aerial nodes is different however, the total aerial nodes to cover the entire ground space is 21 in both the cases. This means that there is no performance improvement by using greedy algorithm II over greedy algorithm I.

8 Dynamic UAV placement algorithm

The objective of the UAV is to cover the environment as rapidly as possible. We first find the likelihood of a target appearing at a particular position on the ground. Based on this likelihood, the UAV moves to view that ground point as quickly as possible. In order to solve this problem, we first divide the whole space into two sets of lattice points, aerial lattice points and ground lattice points. For each aerial lattice point, we find the list of the ground lattice points it can see based on the algorithm described earlier.

If a target appears on the ground, an UAV should try to cover it as rapidly as possible. This problem can be viewed in terms of theory of thermodynamics. If the target appears in a certain area, the temperature in that area increases. An UAV would move towards the target to view it. As soon as an UAV views that target, the temperature of that ground lattice point goes down. The trajectory of an UAV is based on the temperature on the ground and the goal is to follow the path such that at the UAV will cover the highest heated area on the ground.

Assume that a target appears with Poisson process. Probability of a target appearing at ground lattice point z at time t_c is,

$$P(z) = 1 - e^{-\lambda(t_c - t_L)}$$

where t_c is the current time and t_L is the last time the target was seen.

The ground temperature for an aerial lattice point is the sum of the temperature of all the ground lattice points that this aerial lattice point is neighbors with. As described in the previous section, a ground lattice point is called a neighbor with an aerial lattice point if the ground lattice point is covered by that aerial lattice point. Hence, the ground temperature at an aerial lattice point x is defined as:

$$T(x) = \sum_{z} \frac{1}{1 - P(z)}$$

where set z contains all the ground lattice points in that can be seen by the UAV x.

The smooth temperature $F(x)$ is obtained by performing Gaussian smoothing at an aerial lattice point x and it is given by,

$$F(x) = \sum_{z} \frac{1}{1 - P(z)} e^{\frac{-\|x-z\|^2}{2}}$$

The direction in which the UAV should move next is given by,

$$\nabla F(x) = \sum_{y} T(x) \times e^{\frac{-\|x-y\|^2}{2}} \times \begin{bmatrix} x_1 - y_1 \\ x_2 - y_2 \end{bmatrix}$$

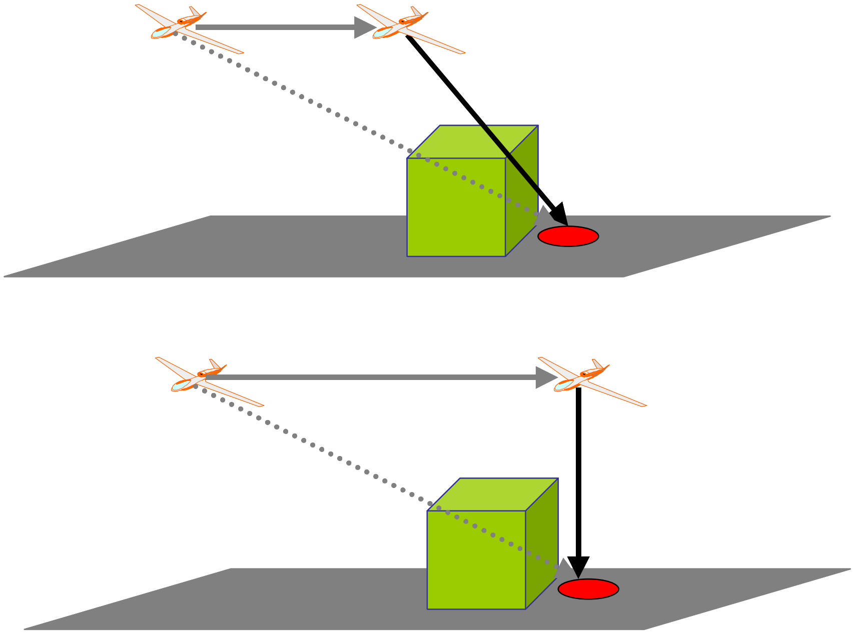

Figure 8.1: The figure shows two different approaches for the dynamic placement of UAV.

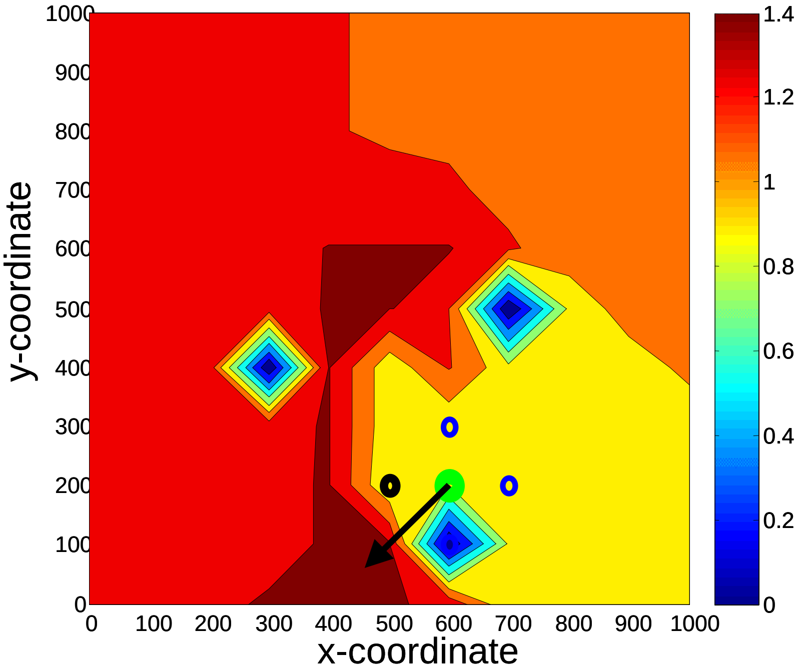

Figure 8.2: A contour plot of $T(x)$. Note that UAV (green circle) moves towards the hottest area (brow area).

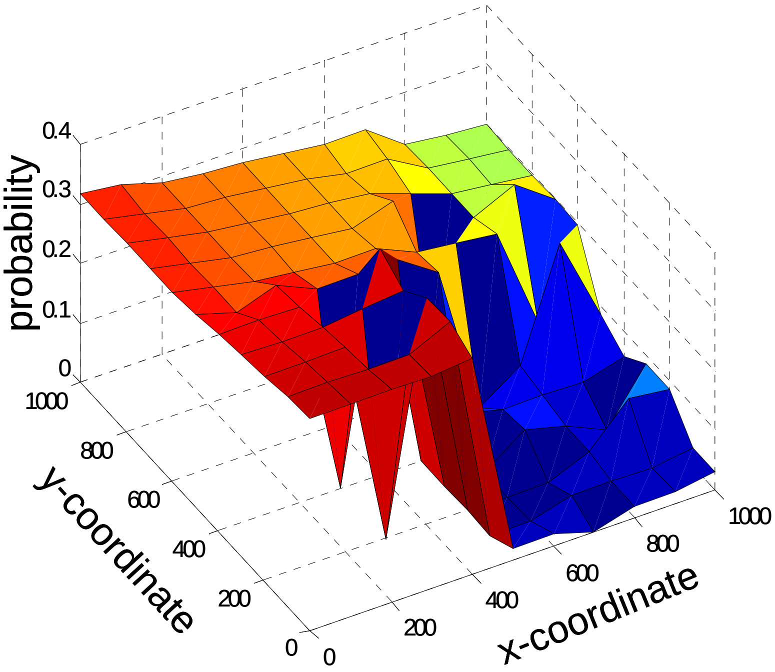

Figure 8.3: The z-axis shows the probability $P(z)$.

where y is the set of all the aerial points except x.

There are two ways to approach this problem:

1. An aerial lattice point moves towards the nearest aerial lattice point that can see the hottest ground lattice point.

2. An aerial lattice point moves to the aerial point that is right above the hottest ground lattice point.

Note that these two approaches are different and that is illustrated in figure 8.1. The upper frame of the figure shows the approach 1, where the aerial lattice point moves to the closest aerial lattice point that can see the hottest spot on the ground. The lower frame of the figure shows the approach 2, in which the aerial lattice point moves to the aerial point that is right above the hottest spot. In both cases, we get different resulting trajectory.

Figure 8.2 shows the contour plot of the $T(x)$ at each ground lattice point. at a specific point in time. The color-bar shows the intensity of the heat. The green circle shows the current position of the aerial lattice point. At the next time step, this aerial lattice point is going to move to its neighbor (the black circle) in the direction of the arrow. This direction is computed from $\nabla F(x)$.In figure 8.2 it can be seen that the aerial lattice point moves toward the hottest ground lattice point (brown area) in terms of $T(x)$. Figure 8.3 shows the 3-D view of the probability $(P(z))$ at each ground lattice point. The red area in the figure shows the highest probability. Meaning that probability of that ground lattice not being seen is very high and the aerial lattice point tries to go there. That can be seen from 8.2. These figures are based on the first approach. The resulting plots for the second approach are not shown here. The performance of this algorithm is evaluated in terms the mean time between viewing a particular ground lattice point. In this example, the resulting MT was 14.72 sec.

9 NMS Traffic Models and Tools

In this section we examine some Models and Tools developed by the DARPA NMS project [22]. These models and tools have been developed for Network Management and Analysis and we need to examine how appropriate they are for the particular case of Military Networks. We are not intending to review all of the tools and approaches developed during the NMS program, but rather to highlight those that appear especially relevant given the new types of problems we have identified in the first part of this report. All of the comments made here are based on what are promising results in available reports and proceedings; any of these promising tools need to be followed up with a more rigorous validation of the claims and a review of necessary refinements needed for future military network scenarios. First we review the several applicable models and approaches. Second, we review simulators and finally, we present a brief discussion on current data capture and analysis tools that could be applied to Military Networks.

9.1 Models and approaches

- Fractional Sum Difference (FSD), to model packet arrivals and sizes on a link, can be used on a Military Network, as its network scale is any link with a traffic rate of more than 4 mbps. The model requires constant mean load so it would not be affected by a peace or conflict period of time.

- IntServ Router with Pricing, to obtain closed loop control of traffic in an IntServ router, only demands the existence of a router and has no limitations on time scale, traffic rate and topology of the network or on the type of data being used so it is applicable to a Military Network.

- DiffServ Router with Pricing, to obtain closed loop control of traffic in an DiffServ router, only demands the existence of a router to model the traffic on the packet level with pricing. Again no other limitations exist, so it can be applied to a Military Network of certain time scale and topology.

- End to end measurement of cross traffic on the bottleneck link of an end to end path, to model this link and also to offer prediction, reconstruction and interpolation can be applied to Military Networks as it also supports multiple bottleneck links thus many different topologies.

- Rapid Model Parameterization (RAMP), to model rapid parameterization of traffic models from real-time networks traffic measurements. It can be used with different traffic rates over a single link, which makes it appropriate for Military Networks. However, RAMP can only model Web traffic and FTP traffic.

- Multi-fractal Wavelet Model (MWM), to model aggregate traffic load on a link, can be applied to Military Networks as it has a wide time scale range and is appropriate for a huge variety of network topologies.
- Alpha-Beta model decomposes the traffic into two-level model, alpha model (Gaussian traffic) and beta model (bursty traffic). This can be used for a variety of network topologies and traffic rates. Thus it can be applied to Military Networks. It can also help to observe how the burstiness of traffic changes as we enter to a conflict period.
- Discrete event (packet-level) models, to model a complete protocol stack, can be applied in a variety of different topologies and can support heavy traffic. It can be used in a Military Networks when discrete events are modeled.
- Moving Picture Experts Group (MPEG) Variable Bit Rate (VBR) Source Traffic, to model I, P and B frames of the MPEG standard according to separate models for the sizes of each type of frame, has no limitations on time scale. It deals with MPEG traffic and can support different traffic rates depending on the setting of parameters. It can be used to a Military Network
- Traffic Models for Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), and Email Applications, which are simple models to capture the main characteristics of application/protocols in MPEG VBR (Variable Bit Rate) model. It can be used in Military Networks, if the network topologies that were used to validate these models are similar to the Military topologies.
- Fluid models for congestion controls, to model the dynamics of congestion control and active queue management mechanisms on the Internet, is applicable to long-lived file transfers which is not always the case in Military Networks – unless e.g. a backup or update of a database. However, the model is independent of network topology or traffic rate so it can be used in Military Networks if the time scale conditions hold (min up to hours).
  - Fluid Flow models with an interface to a packet level simulator, to model the behavior of TCP with RED AQM, has no limitations on the network size. However it currently works with Static Routing which is often but not always the case in Military Networks. Also the validation of this model has be made in cases where the amount of traffic is significantly large. This is not always the case for Military Networks especially in peace periods of time.

  - Fluid Model for IEEE 802.11 Distributed Coordination Function (DCF) models the transmission activities under the DCF for a IEEE 802.11 wireless LAN and is applicable to any time scale and different network topologies and traffic rates so it can be applied to a Wireless Network. However, it should be noted that the error ranges up to 2% of the system throughput value and this may be not small enough for a Military Network, especially for conflict periods of time with increased traffic load on the network.
  - Discrete-Event Fluid Model of TCP, to model TCP protocol, is valid in network topologies as the ones used during the study – we don't know if this is similar to a Military Network so we can't say how appropriate it would be. It also demands TCP traffic. It has no time scale limitations.

9.2 Simulators

In order to validate any of the above network models and approaches, NMS program recognized that we needed much better ways to simulate existing and developing networks, and much better ways of analyzing based on such simulations. Here, we list several simulators that can be useful for simulating military networks.

- HNS Hybrid Network Simulation, that models the movement of workload in a network, can be used for any time scale (time horizon) so it won't have a problem to be applied in a Military Network. However, the traffic rate should be in the order of 1000 flows so this may create a problem in periods where the Military Network faces totally different traffic rates.

- JavaSim, which is a component-based, compositional network simulation, emulation, and protocol synthesis environment. It can be used with Military Networks.

- Genesis Network Simulator (NS-2), which is a tool that enables packet-level, fast, computation distributed network simulation under NS2 using Genesis fixed-point-solution approach. The speedup in simulation time again may lead to higher error percentage that may not be acceptable for Military Networks. If this is not a problem it can be used with Military Networks.

- Genesis- GloMoSim, which is a tool to simulate packet-level fast, computation distributed network simulation. No additional requirement – C++ code. It can be used on a Military Network. Network Daemons for Low End-to-End message delay (NetLets), which is a tool to implement an overlay network of in-situ instruments on wide area networks. It's a stand-alone application that can be used with a Military Network.

- The Georgia Tech Network Simulator (GTNetS), which is a tool to be used where existing sims lack capabilities. It can be used with Military Networks.

9.3 Tools for data capture and analysis

In order to validate the analytical network models and approaches, NMS program noticed that we needed much better ways to capture real data from current and developing networks, and much better ways of analyzing such data. Hence NMS developed and utilized a rich array of data collection and analysis techniques. For example, Centralized Secure collection of data from multiple sites that coordinates the collection and storage of traffic measurements from probes at multiple sites can be extremely useful in Military Networks where centralized control is one of the main characteristics. Tools such as Domain Name System Statistics (dnsstat), Network Traffic Measurement Tool (NeTraMet), and “Veil” emulation system can be highly applicable in capturing the necessary traffic. For real-time monitoring of global Internet routing in military networks Renesys is remarkably relevant. Many tools like CoralReef, Self-similarity analysis software (SELFIS), Asim are immediately useful for data analysis or seem to be easily adaptable for analyzing network traffic. Others like Multiparadigm network modeling framework (Maya) and WQRSP (Wireless QoS Routing and Scheduling Protocol) are clearly useful to some of the future scenarios we have dealt with in this report.

10 Conclusion

Current NMS models and approaches based on commercial networks are clearly relevant to military needs. However, military networks differs from commercial networks in some characteristics. Such characteristics include the differences in priorities and loads during military events vs. peacetime or normal operation, the special problems of malicious disruption or co-opting of the network for enemy uses, and the fact that some network routes may be fully controlled, prioritized, and knowable (in other words not characterized best by statistical models). Hence, we believe that there is also a need to have many specialized methods for military modeling of and control of networks, especially for military network needs during times of crisis and real operations.

Based on this need, this study originally sought to identify existing military network datasets that could be cleaned up and used for NMS projects. Unfortunately, when many existing datasets were actually examined their own authors reported their weaknesses. One of the chief weaknesses was their lack of information on some of the new characteristics of networks, e.g. mobile sensor-based etc. with this in mind, we refocused this study to concentrate on what are some of the new models needed for realistic reasoning about future networking. We then conclude the study with a brief look at the NMS projects and models and highlight those that appear to be especially useful to these future scenarios.

Reference List

[1] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Mobile Computing and Networking*, 1998, pp. 85–97.

[2] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *WCMC*, vol. 2, no. 5, pp. 483–502, 2002.

[3] V. Tolety, "Load reduction in ad hoc networks using mobile servers," Master's thesis, Colorado School of Mines, 1999., 1999.

[4] F. Bai, N. Sadagopan, and A. Helmy, "Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks," in *InfoCom*, 2003.

[5] Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri, "Towards realistic mobility models for mobile ad hoc networks," in *MobiCom*, 2003.

[6] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang, "A group mobility model for ad hoc wireless networks," in *Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Syste*, 1999, vol. 2.

[7] Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, and Mikael Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," in *MobiCom*, 1999.

[8] I. Rubin and R. Zhang, "Performance behaviour of unmanned vehicle aided mobile backbone based wireless ad hoc networks," .

[9] S. Benjamin and I.Rubin, "Connected disc covering and applications to mobile gateway placement in ad hoc networks," *Proceedings of ADHOC NetwOrks and Wireless (ADHOC-NOW)*.

[10] D.L. Gu Et. al, "Uav guided intelligent routing for ad-hoc wireless networks in single-area theatre," .

[11] N. Megiddo and K. J. Supowit, "On the complexity of some common geometric location problems," *SIAM Journal on computing*, pp. 13, 182–196, 1984.

[12] He Computational Complexity of the M-Center Problems on the Plane, ," *Transactions of the IECE of Japan*, vol. E64, pp. 57–64, 2002.

[13] R. Chen, "Solution of minisum and minimax location-allocation problems with euclidean distances," *Naval Research Logistics*, 1983.

[14] Z. Drezner, "The p-center problem -heuristic and optimal algorithms," *Journal of the Operational Research Society*, 1984.

[15] R. Vijay, "An algorithm for the p-center problem in the plane," *Transportation Science*, 1985.

[16] R. Chen and G. Y. Handler, "A relaxation method for the solution of the minimax location-allocation problem in euclidean space," *Naval Research Logistics Quarterly*, 1987.

[17] Karthikeyan Chandrashekar Majid Raissi-Dehkordi and John S. Baras, "UAV placement for enhanced connectivity in wireless ad-hoc networks," *technical report*, 2004.

[18] Prithwish Basu, Jason Redi, and Vladimir Shurbanov, "Coordinated flocking of UAVs for improved connectivity of mobile ground nodes," *Milcom*, 2004.

[19] Scalable Network Technologies, "The QualNet simulator http://www.qualnet.com/," .

[20] "Geographic information system (GIS), http:
www.gis.com," .

[21] http://infosun.fmi.uni-passau.de/Graphlet/GML/, ," .

[22] http://www.nms-integration.org/, ," .